



Git e Github

Módulo 3 - Colaboração na prática

Onde estão os textos
da tabela?

Retomando o que ficou pra trás..

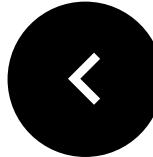


SALTANDO A STAGING ÁREA

- Posso fazer para todos os arquivos? Novos e antigos?
- Para onde o marcador HEAD aponta?

2021

Retomando o que ficou pra trás..



SALTANDO A STAGING ÁREA

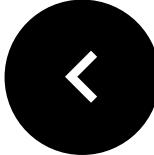
- Posso fazer para todos os arquivos? Novos e antigos?
- Para onde o marcador HEAD aponta?

OUTROS COMANDOS DO GIT

- Como faço para buscar mais informações sobre o histórico de commits?
- Qual a função de um arquivo .gitignore? Como ele é formatado?

2021

Retomando o que ficou pra trás..



SALTANDO A STAGING ÁREA

- Posso fazer para todos os arquivos? Novos e antigos?
- Para onde o marcador HEAD aponta?

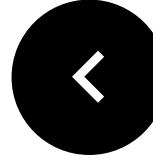
OUTROS COMANDOS DO GIT

- Como faço para buscar mais informações sobre o histórico de commits?
- Qual a função de um arquivo .gitignore? Como ele é formatado?

DESFAZENDO AS COISAS

- Como removo todas as modificações que fiz em um repositório ainda fora da staging area? E quando já adicionado? E quando já commitado?
- O que é fazer um amended commit (commit emendado) ? Quando NÃO posso fazer-lo?
- Como reverter uma mudança? O marcador HEAD tem alguma aplicação nesses casos? Como faço para reverter para um commit específico?

Retomando o que ficou pra trás..



SALTANDO A STAGING ÁREA

- Posso fazer para todos os arquivos? Novos e antigos?
- Para onde o marcador HEAD aponta?

OUTROS COMANDOS DO GIT

- Como faço para buscar mais informações sobre o histórico de commits?
- Qual a função de um arquivo .gitignore? Como ele é formatado?

DESFAZENDO AS COISAS

- Como removo todas as modificações que fiz em um repositório ainda fora da staging area? E quando já adicionado? E quando já commitado?
- O que é fazer um amended commit (commit emendado) ? Quando NÃO posso fazer-lo?
- Como reverter uma mudança? O marcador HEAD tem alguma aplicação nesses casos? Como faço para reverter para um commit específico?

RAMIFICAÇÕES E JUNÇÕES

- O que é uma branch? Qual a principal vantagem de usar-la?
- O que é um merge? Como corrigir problemas nele?
- Quais os tipos de algoritmos usados no merge?



Agenda de Hoje

- Entender o Github
- Repositórios remotos
- Colaboração
- Exercícios
- Boas práticas

Afina de contas, o que é o Github?



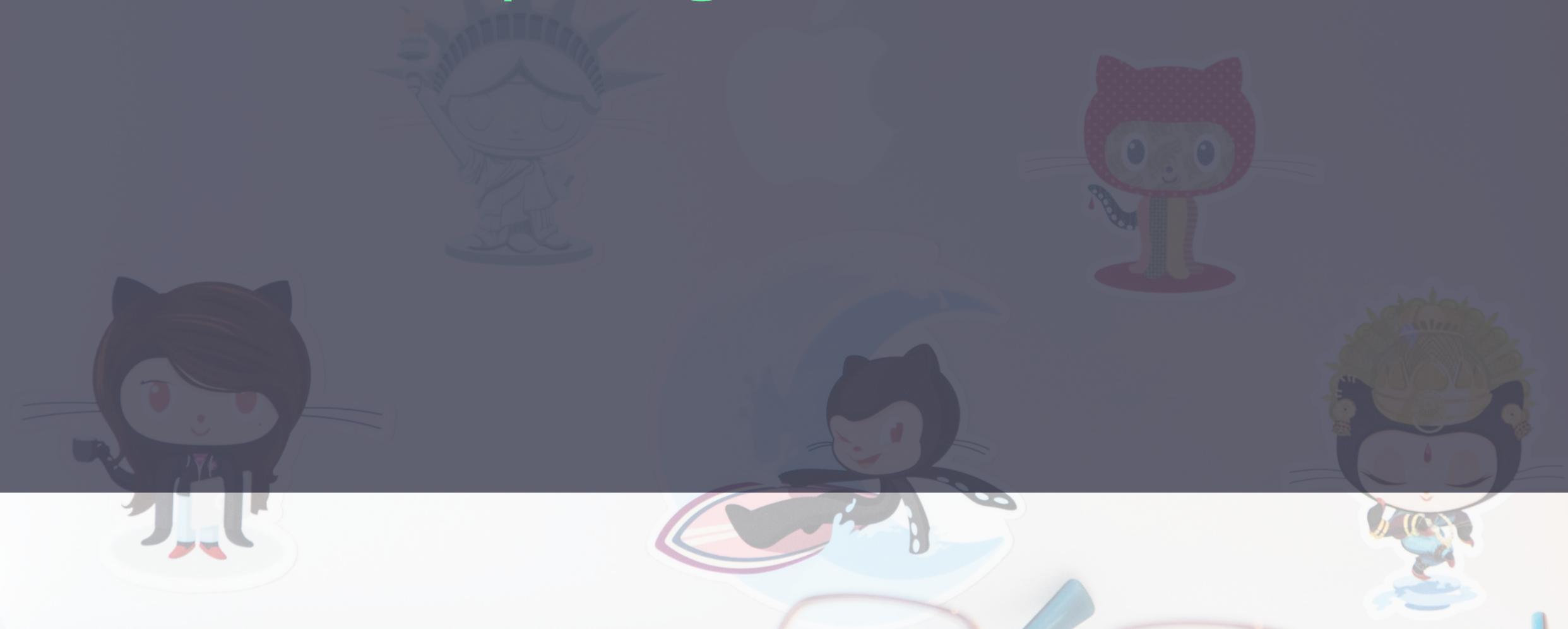
Make your contribution



Github

Interface web para o protocolo git, provendo servidores de repositórios remotos (públicos ou privados) além de outras ferramentas de gestão de projetos, equipes e permissões.

<https://github.com>



git clone



Diferenças entre Download e Clone



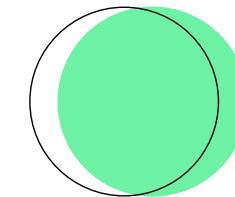
git push



Preciso ficar
colocando usuário e
token toda hora?

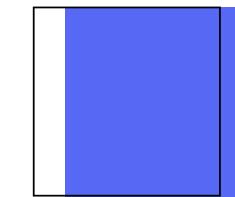


Preciso ficar colocando usuário e token toda hora?



SSH KEYPAIR

Adiciona no Github a sua chave pública de ssh para que ele saiba que você é você mesmo



CREDENTIAL HELPER

Faz um cache de 15 minutos das suas credenciais para que você não precise ficar repetidamente digitando usuário e token.

git config --global credential.helper cache

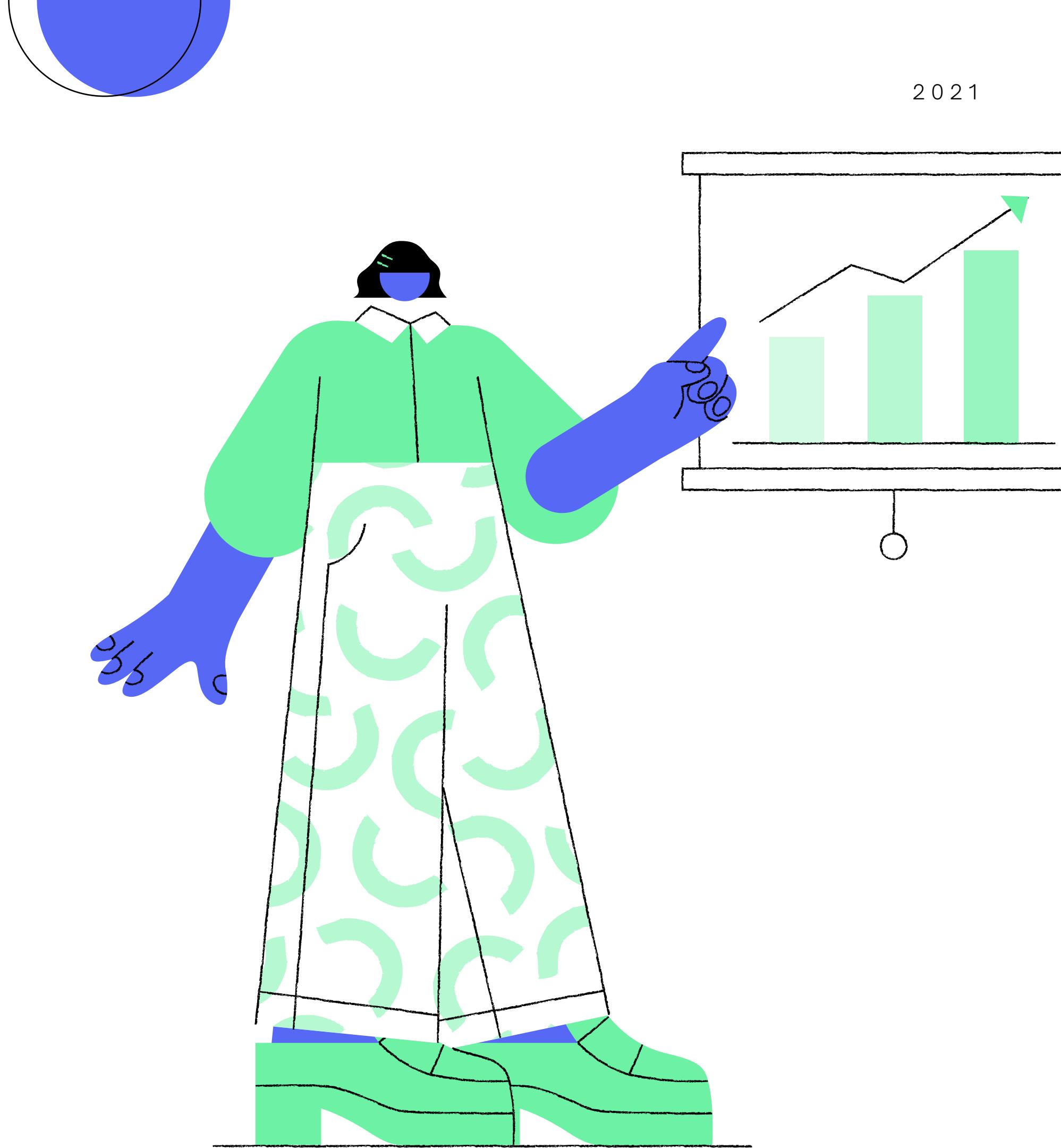


git pull



Repositórios Remotos

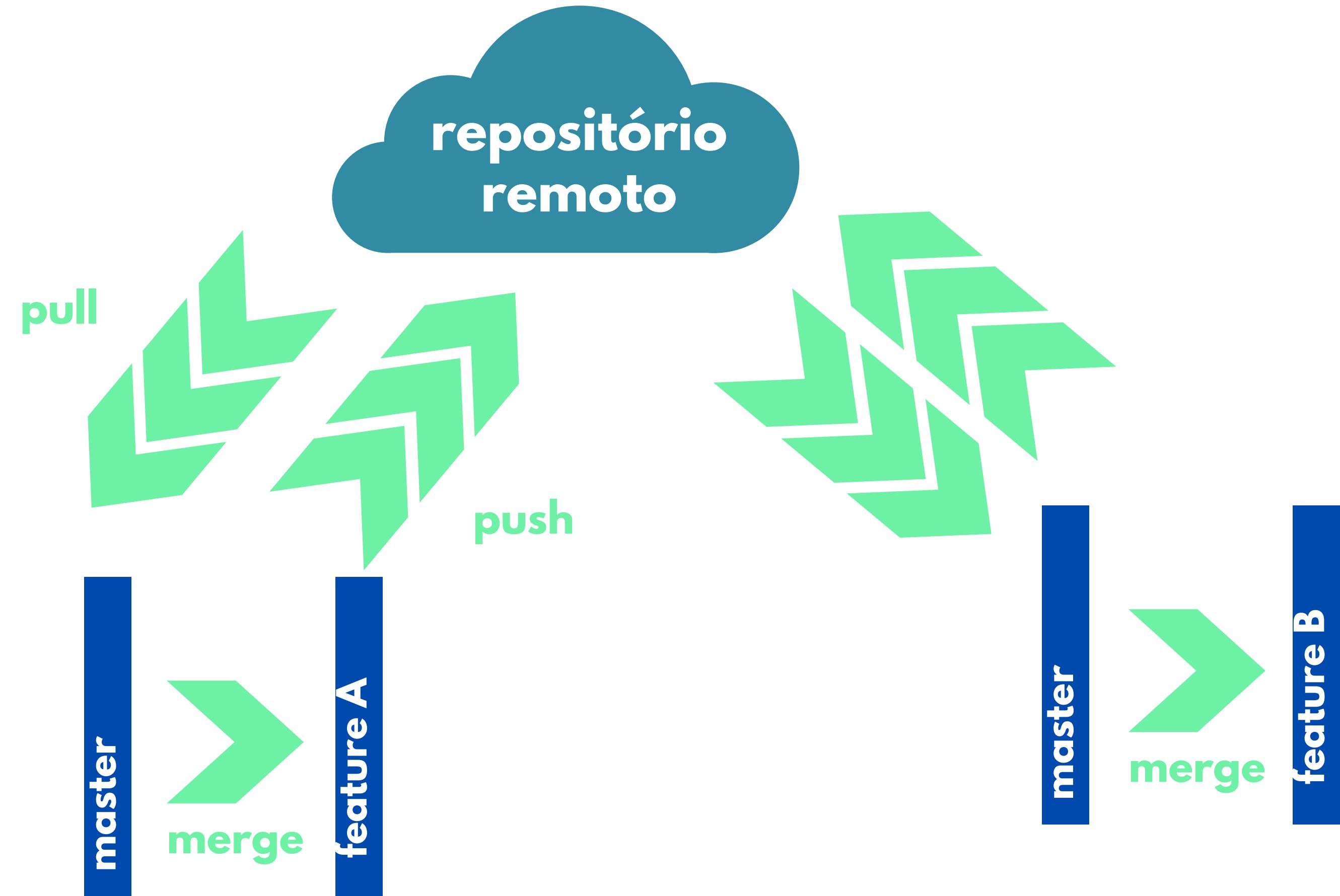
2021



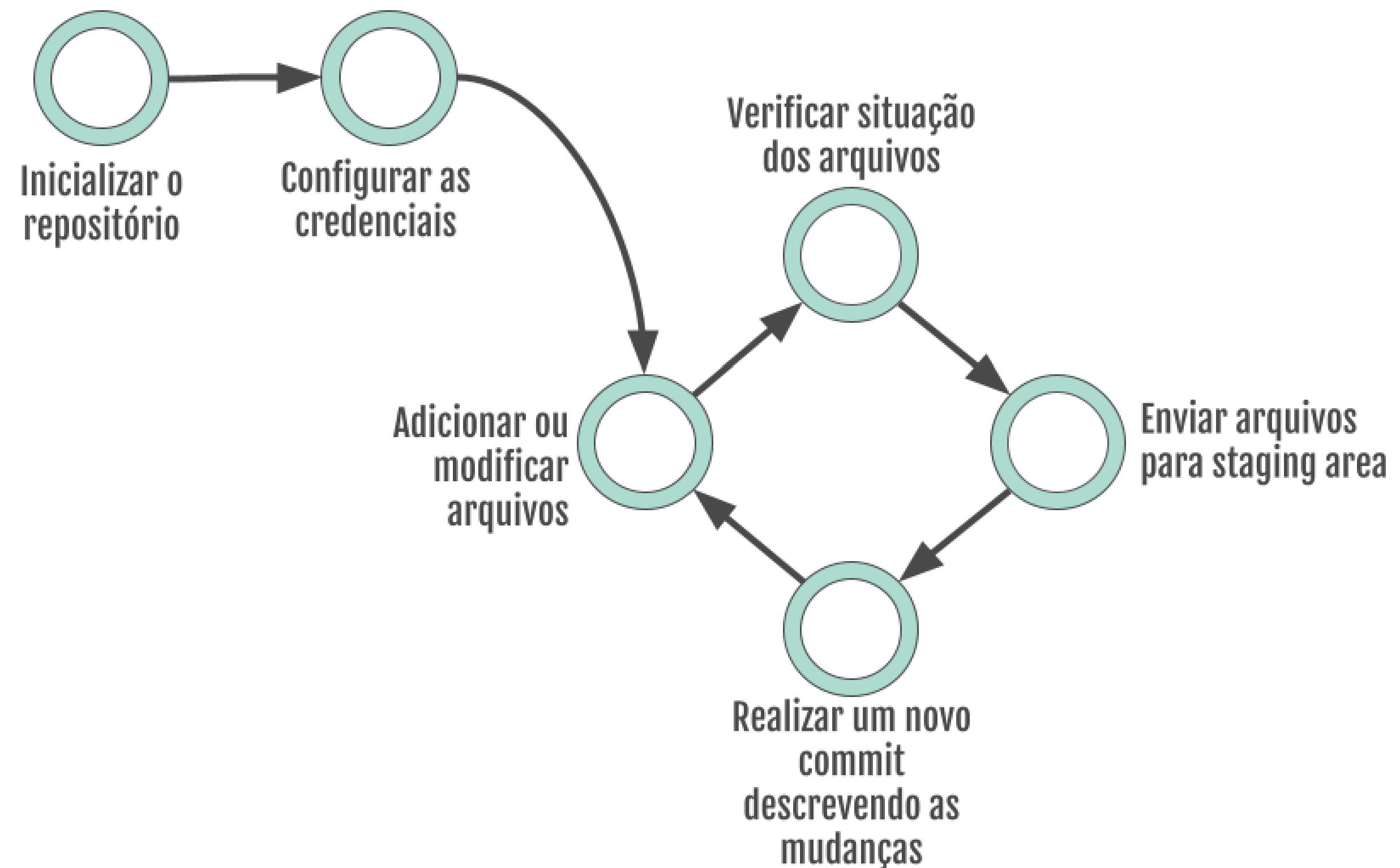
Alternativas ao Github



**Google Cloud
Source Repositories**



git workflow



git workflow



git remote -v



git remote show origin



git fetch



**Conflitos...
Bora resolver !**



Enviando branches para o repositório remoto



Gestão de versões em diferentes branches

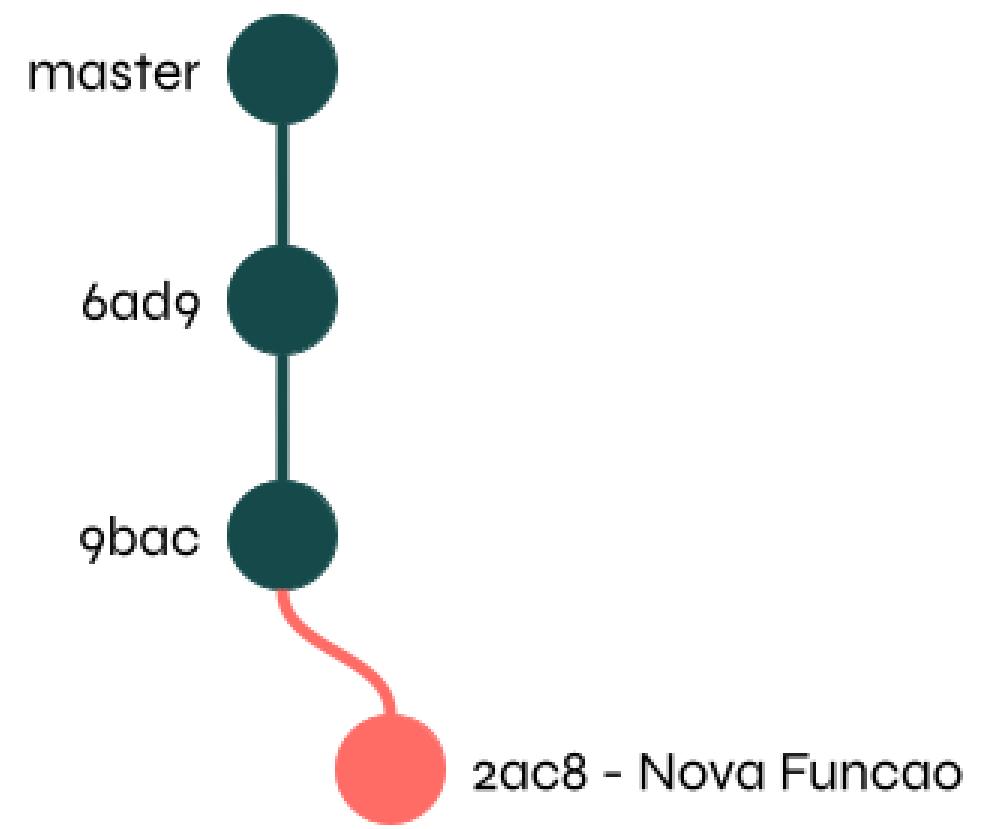
- ISSO É UMA POLÍTICA DO PROJETO
- PROJETOS PODEM MANTER O HISTÓRICO LINEAR OU RAMIFICADO, NUNCA OS DOIS
- HISTÓRICO LINEAR É MELHOR PARA LOCALIZAR FALHAS
- HABITUALMENTE, NA BRANCH PRINCIPAL (MASTER) FICA A PRÓXIMA VERSÃO QUE IRÁ PARA PRODUÇÃO
- EM UMA OUTRA BRANCH, FICA A VERSÃO ATUAL DO CÓDIGO.

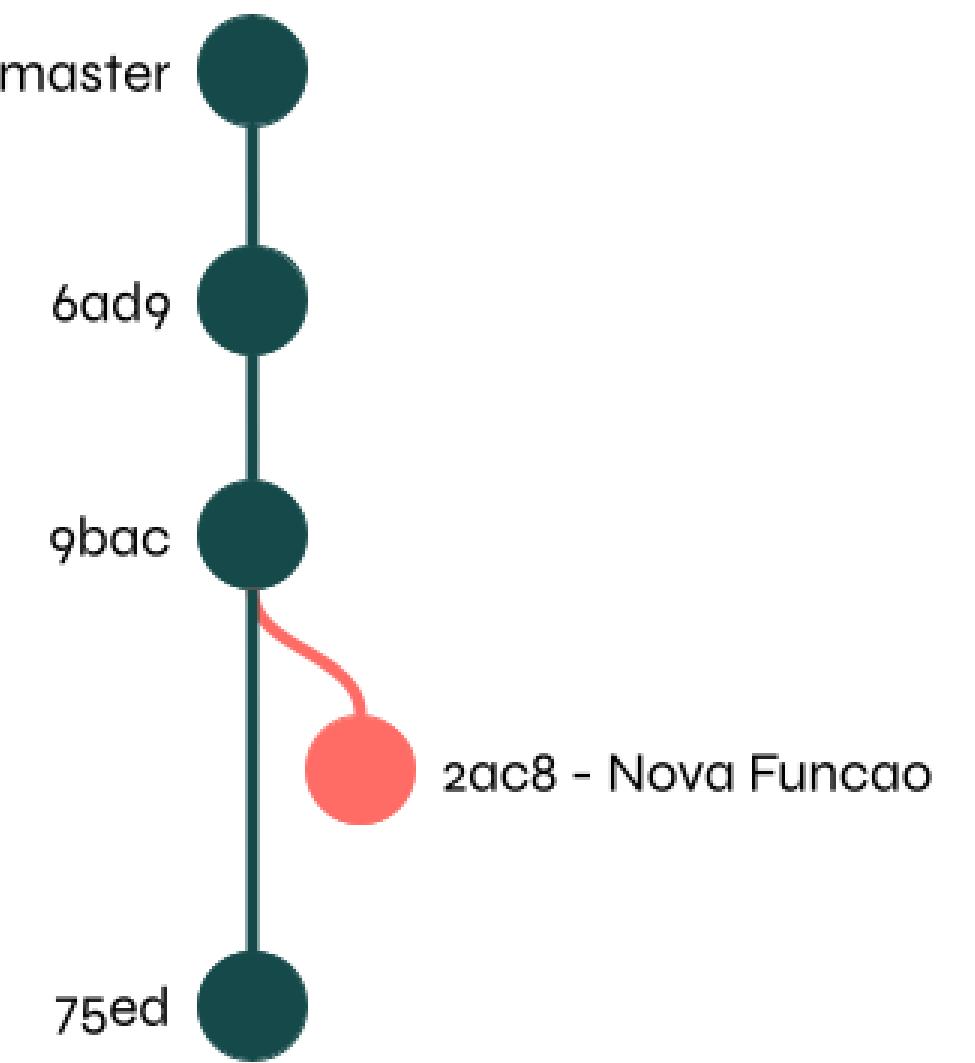
```
$ git checkout -b newBranch  
$ git add .  
$ git commit -m "..."  
$ git push -u origin newBranch
```

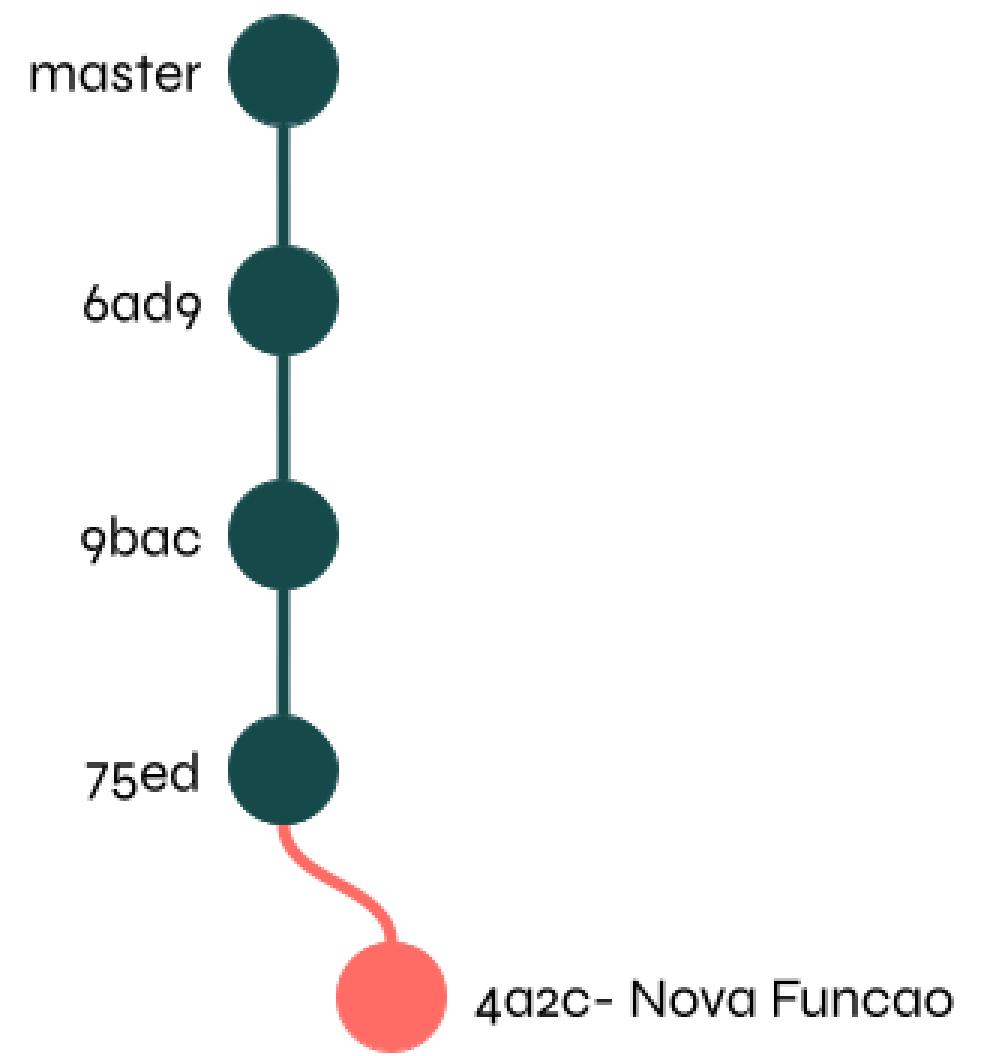


**REBASE ... mas com
cuidado**





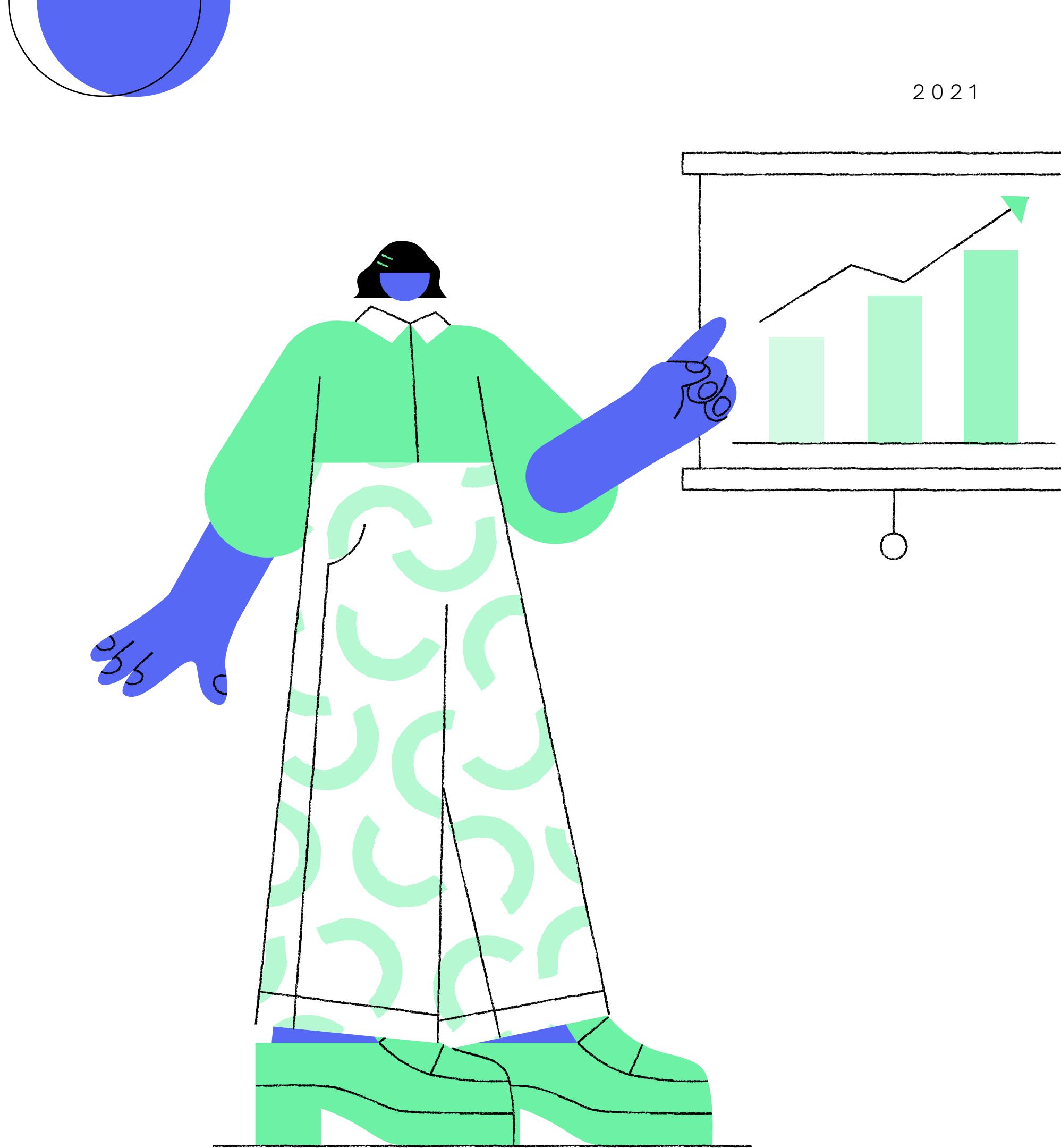




```
$ git checkout newBranch  
$ git rebase master
```



Boas práticas



Boas práticas na gestão de repositórios

SINCRONIZE SEU TRABALHO ANTES DE COMEÇAR

Antes de iniciar qualquer grupo de tarefas, sincronize com o repositório remoto. Assim, você não precisará resolver grandes blocos de conflitos na hora do merge

Boas práticas na gestão de repositórios

SINCRONIZE SEU TRABALHO ANTES DE COMEÇAR

Antes de iniciar qualquer grupo de tarefas, sincronize com o repositório remoto. Assim, você não precisará resolver grandes blocos de conflitos na hora do merge.

EVITE MUDAR MUITA COISA DE UMA SÓ VEZ

Fez uma refatoração e mudou o nome de uma variável, faça o commit. Assim, caso algo dê errado, fica mais fácil saber onde voltar.

Boas práticas na gestão de repositórios

SINCRONIZE SEU TRABALHO ANTES DE COMEÇAR

Antes de iniciar qualquer grupo de tarefas, sincronize com o repositório remoto. Assim, você não precisará resolver grandes blocos de conflitos na hora do merge.

EVITE MUDAR MUITA COISA DE UMA SÓ VEZ

Fez uma refatoração e mudou o nome de uma variável, faça o commit. Assim, caso algo dê errado, fica mais fácil saber onde voltar.

QUANDO FOR TRABALHAR EM UMA FUNÇÃO GRANDE, FAÇA EM UMA BRANCH SEPARADA

Isso irá reduzir drasticamente a chance de conflitos com outras partes do seu projeto.

Boas práticas na gestão de repositórios

FAÇA O MERGE DAS SUAS BRANCHES COM A MASTER FREQUENTEMENTE

Quando trabalhamos com muitas branches em paralelo, ficar pra trás no tempo é muito fácil e muito perigoso.

Boas práticas na gestão de repositórios

FAÇA O MERGE DAS SUAS BRANCHES COM A MASTER FREQUENTEMENTE

Quando trabalhamos com muitas branches em paralelo, ficar pra trás no tempo é muito fácil e muito perigoso.

MANTENHA A VERSÃO ATUAL DO PROJETO EM UMA BRANCH DIFERENTE DA VERSÃO BETA

Caso qualquer coisa aconteça com seu projeto na hora de integrar as novas funções, assim fica muito mais fácil voltar para quando tudo funcionava... :-D

Boas práticas na gestão de repositórios

FAÇA O MERGE DAS SUAS BRANCHES COM A MASTER FREQUENTEMENTE

Quando trabalhamos com muitas branches em paralelo, ficar pra trás no tempo é muito fácil e muito perigoso.

MANTENHA A VERSÃO ATUAL DO PROJETO EM UMA BRANCH DIFERENTE DA VERSÃO BETA

Caso qualquer coisa aconteça com seu projeto na hora de integrar as novas funções, assim fica muito mais fácil voltar para quando tudo funcionava... :-D

NADA DE REBASE EM REPOSITÓRIOS COMPARTILHADOS

O rebase muda a história do projeto. Por si só, isso é um grande efeito colateral.

Boas práticas na gestão de repositórios

FAÇA BOAS MENSAGENS DE
COMMIT

RESOLVA CONFLITOS PARTE A
PARTE. NUNCA DEIXE PARA SUA IDE
RESOLVER TUDO.

Obrigado!

Agora é hora de ir pra casa... :-D

